

HOW IT WORKS

JPEG encoding

Uncompressed, the average digital photo would take up many megabytes of space on your PC's hard disk. Simon Handby looks at how JPEG compression squeezes photos down to a more manageable size

We're spoiled by the number of high-quality digital photos we can store on our PCs, and the sheer number of images that adorn webpages all over the internet. But have you ever stopped to wonder how these images are stored? Without compression, even the simplest image would take up a few megabytes of storage space and take a while to download even over a broadband connection. The Joint Photographics Expert Group (JPEG) compression standard can reduce an image file's size by up to 20 times, and is by far the most popular image format; it's used by computers, cameras and websites all across the internet. This month, we'll explain how the technology works.

BULK OF THE PROBLEM

The bitmap images displayed on a computer monitor are made up of many thousands of individual pixels, each of which can be a different brightness and colour. These days, most LCD computer screens run at 1,280x1,024 SXGA resolution, giving a total of more than 1.3 million pixels. The overall colour of each consists of a

red, blue and green component, each of which can be any one of 256 values.

Storing the value of each colour requires an 8-bit binary number ($2^8=256$), so each pixel requires 24 bits of information. With so many pixels on the screen, storing the entire image requires 3.75MB. Modern computers can display and manipulate full-screen and much larger images with ease, but it's often useful to make them smaller before storing or transmitting them.

For data files such as spreadsheets and databases, information must be stored and retrieved without being altered. If the file needs to be compressed, the process must be 'lossless', otherwise its data will become corrupted. Consequently, all general file compression formats such as ZIP and Windows XP's own file and folder compression are lossless.

When used to compress a multimedia file such as an image or audio track, lossless compression preserves information perfectly, but it generally doesn't reduce the file size by more than a few per cent. However, it's possible to remove plenty of information from most

multimedia files without any noticeable loss of detail, leading to a much smaller file.

Such compression is known as 'lossy' (see *How it Works*, December 2005), and key to the process is the idea that data that will be missed least is removed first. By allowing a slight loss of fidelity during compression, lossy formats such as JPEG and MP3 can typically reduce the storage space for a multimedia file by 10-20 times.

THE GREAT COMPRESSION

The JPEG standard allows for some flexibility, but most applications create JPEG File Interchange Format (JFIF) files when saving an image as a JPEG. This involves six main steps, the first of which is to convert the red, green and blue (RGB) information for each pixel in the image into values for brightness, blue and red.

This step doesn't remove any information, as the green values can be re-calculated from the other three, but it allows any subsequent compression to treat brightness and colour information separately. Because the human eye is more sensitive to changes in brightness than to subtle differences in colour, it's possible to apply more aggressive compression to the colour information while preserving more brightness detail. As a result, you get the same perceived image quality from a smaller file.

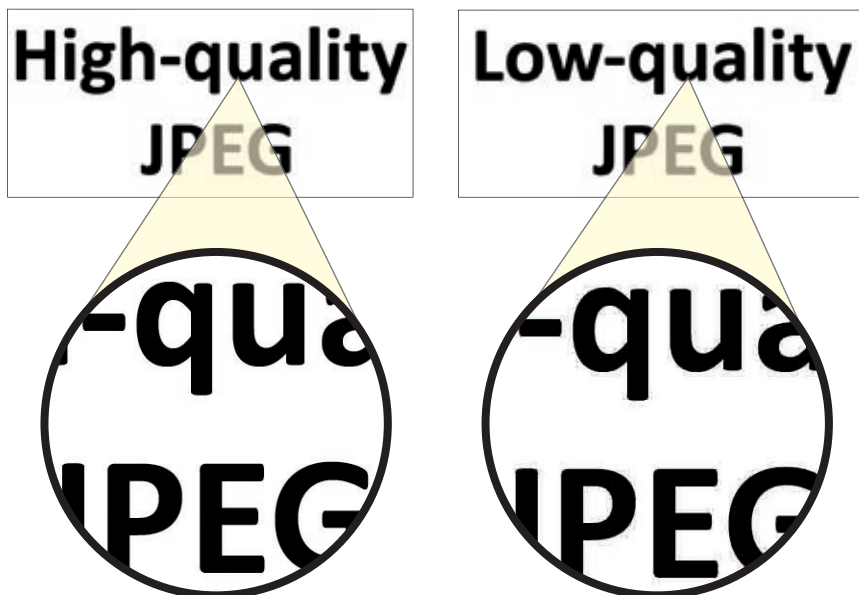
With the colour space transformed in this way, the encoder can now reduce or 'downsample' the distinct levels of blue and red contained within the image. It does this by averaging the values between two horizontally neighbouring pixels, and often by also averaging between vertical neighbours. As colour information accounts for two-thirds of the information in an image, halving it by averaging horizontally neighbouring pixels reduces the total image file size by a third. Reducing it by three quarters, by also averaging vertically, halves the overall image file size.

The third step in creating a JPEG is to split the image into blocks eight pixels high by eight pixels across (64 pixels in total). Few images' dimensions are an exact multiple of eight, so at this stage the encoder usually needs to add some dummy data. A common technique for this is to repeat the pixels at the edge of the image to fill the extra space. Other encoders add pixels that are the average colour and brightness of the existing pixels in the block.

The information in each block is now processed individually so that it takes up less storage space. This works in a similar way to the encoding of audio information in an MP3 file, where the data in small blocks of audio is analysed using a mathematical function known

Precious artefacts JPEG compression in action

These images show the difference between high- and low-quality JPEG compression, with the highest quality on the left and the lowest quality on the right. At the lowest quality settings, problems are easy to spot and artefacts (the blocky appearance around the text) are clearly visible. These problems come about because the compression throws away too much information in each 8x8 pixel block. The problem is worse with high-contrast images, such as those containing text, but can be visible in photographs, too.



as a Fourier transform. This function allows complex wave information, such as a sound file, to be expressed as a set of simple sine waves with different frequencies and strengths.

For a music file, you can think of the procedure as similar to the way a spectrum analyser's bars show the relative volume of bass, mid-range and treble in a song. If the encoder analyses and records the strength of enough frequency bands, it can reproduce the original audio signal by combining sine waves of the correct frequencies and strengths.

The process of creating a JPEG is slightly more complex as it involves values for brightness, blue and red for pixels arranged in two dimensions. A discrete cosine transform (DCT), similar to a Fourier transform, analyses the distribution of the values within the block and represents them as frequency data. In this data, high frequencies represent rapid variations in values between pixels, and low frequencies show changes that are spread over a wider area. After the DCT function has been applied, we're left with 64 frequency values per block. At this point in the process we can reverse the steps and re-create the image exactly.

QUANTISATION

Although the eye can spot subtle differences in brightness over a relatively large area, it's less good at picking out the precise details of tightly packed brightness variations. As a result, once the information from the image is represented as frequency data, it's possible to reduce the amount of high-frequency detail massively. In fact, many of the higher-frequency components can be discarded altogether.

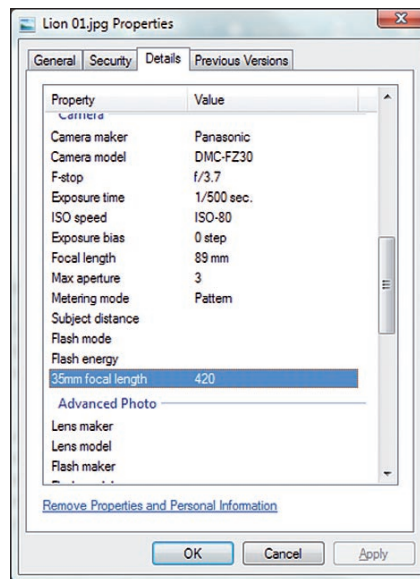
By dividing each of the 64 frequency values by a constant and then rounding it to a whole number, each becomes smaller and requires less storage space. If the quantisation constant were six, for example, and there were four frequency values, 23, 2, 44 and 41, they would be stored as 4, 0, 7 and 7, each of which can be represented in three bits of data or less. This is the 'lossy' part of the compression, because once we've thrown data away we can't re-create it exactly.

Although quantising the values in this way is the single most effective step in reducing the file size of a JPEG image, it also means discarding a great deal of information. When the image is reconstructed again, our example quantised values above would be multiplied by six to give 24, 0, 42 and 42, a significant difference, and therefore loss of fidelity, from the originals.

In practice, compression isn't so crude as to use just one constant for the entire image. Instead, each of the 64 frequency components is divided by a constant that is selected for that frequency, so low-frequency components (which are more easily seen by the eye) are generally preserved with more detail. The constants used are contained within a matrix of 64 values that determine the overall compression ratio and quality of the JPEG image. Different encoders apply a different quantisation matrix depending on the quality setting (which you select as a quality percentage when you save a JPEG in your photo-editing software). The matrix you use is saved with the JPEG so that the decoding application knows which values to use.

NO GREAT LOSS

Lossless compression is applied to the data after the previous four steps. In most cases the



▲ JPEGs store EXIF information about resolution, quantisation and even the camera that made them

encoder does this sequentially, encoding the data from each block in turn to produce the final information, but the JPEG standard also supports progressive encoding.

In this case, the encoder makes several passes over the data. In the first pass, it encodes data from the same position in each block, then it adds data from the next position in each block and so on. If you view a file encoded in this way over a slow internet connection, for example, you can quickly see a crude representation of the final image, and further detail is added as your browser receives and decodes each successive scan.

The final step in creating a JPEG is to package the encoded image within a file header that contains information such as the quantisation matrix and image resolution, and which may also include details of the software or camera that created the file. You can view this exchangeable image file format (EXIF) information in most image-editing programs by choosing the 'file info' or similar option. It's also displayed by photo-sharing websites such as www.flickr.com.

A MATTER OF ARTEFACT

Before a JPEG image can be displayed, it must be decoded in a reversal of the encoding process. First, the image data is extracted from the file header and expanded to reverse the lossless compression. By multiplying the quantised frequency information using figures from the quantisation matrix stored in the file header, the decoder can restore it to its approximate original values.

Next, the decoder performs an inverse DCT on the frequency data for each 8x8 block, reproducing the original values for brightness, blue and red in turn. The blocks are then recombined to form a whole image. By referring to the original image size, the decoder can remove any dummy pixels that were added at the picture's edges. Finally, values of brightness, blue and red are converted back to RGB, and the image can be displayed.

Because some information was thrown away when making the JPEG, the extracted image does not match the original exactly. At typical settings, though, it's hard or impossible to spot the difference. At low quality settings, which produce the greatest degree of compression, you may be able to see blocky artefacts, particularly around areas of high contrast. For this reason, JPEG compression isn't ideal for text or line art, but it excels at compressing photographs.

This strength makes JPEG ideally suited to consumer digital cameras, which often take large images that, uncompressed, would require several megabytes each to store. A typical 5-megapixel camera might produce a 14MB image, but even high-quality JPEG compression can reduce this to 2MB or less, allowing the camera to hold seven times as many images and to write to its memory card more quickly. One problem with the encoding, however, is that such large images take a fair amount of processing, adding to the drain on the camera's battery.

The loss of quality in even the best JPEGs makes them unsuitable for professional photography, so most digital SLR cameras can save unprocessed RAW image data. Apart from some basic operations, such as a 90° rotation, a JPEG must be decoded before the image can be edited. Saving the image again loses even more information, so it's best to use a lossless format such as TIFF until any editing is finished.

FUTURE PROOF?

Despite its continued widespread use, JPEG has technically been superseded by a newer format: JPEG 2000. The Joint Experts Photographic Group developed this format in 2000 to address the shortcomings of the original JPEG standard. JPEG 2000 produces similar file sizes for a given quality, but instead of JPEG's characteristic blocky artefacts at high compression ratios the image becomes subtly blurred, which can be less noticeable.

JPEG 2000 has never achieved widespread use, but its prospects improved in 2002 when Forgent Networks claimed that it owned a patent on the compression technology used in the original JPEG standard, but which isn't used in JPEG 2000. Although some companies using JPEG technology chose to pay a licence fee, Forgent's claims were eventually broadly rejected by the US Patents and Trademark Office. JPEG 2000 remains comparatively unsupported and rarely used.

The Joint Photographic Experts Group recently announced that a new JPEG standard designed for the next generation of digital cameras is under consideration. The format, currently called HD Photo, uses Microsoft-patented technology, although the company has agreed to provide it free of charge. According to the HD Photo specification, the new format "offers image quality comparable to JPEG 2000 with computational and memory performance more closely comparable to JPEG".

FURTHER INFORMATION

Wikipedia entry on bitmap (raster) graphics:

http://en.wikipedia.org/wiki/Raster_graphics

Joint Photographic Experts Group homepage:

www.jpeg.org

Wikipedia entry on quantisation:

www.tinyurl.com/652gh

Compressed image formats compared:

www.tinyurl.com/2em64u